B2 C1

10. (Amended) A method for processing software instructions comprising:

providing two microinstructions to emulate a high-half and a low-half SSE operation,

forcing the high-half and low-half operations to issue in parallel,

dispatching the high-half and low-half operations simultaneously to a first FP unit and to a second FP unit, respectively,

executing the high-half and low-half operations simultaneously, in lockstep,

generating a signal from an emulator's hardware,

sending the signal to the first and second FP functional units,

determining whether an exception is taken in either the first or the second FP unit,

if an exception is taken in either the first or second FP unit,

preventing results from the high-half and low-half operations from being written to result registers, and

canceling both the high-half and low-half operations, and

updating MXCSR flags based upon the results of the first and second FP units.

B3 C1

12. (Twice Amended) A computer system comprising:

a processor comprising,

a floating point unit comprising a plurality of functional units adapted to execute microinstructions;

a ROM;

a plurality of floating point registers;

wherein the processor is configured to emulate an instruction set by:

decomposing a macroinstruction into a plurality of microinstructions;

issuing all of the plurality of microinstructions simultaneously, in parallel, to the functional units,

determining whether an exception occurs in any of the functional units,

setting result registers for results of each of the functional units only if no exception occurs in any of the functional units, and

if an exception occurs in any of the microinstructions, canceling all of the microinstructions and preventing the setting of result registers for all of the functional units.

Please add the following new claims 21-22:

B4 C1

--21. (New) The method of claim 1,

wherein the step of executing comprises executing using a plurality of functional units of a floating point unit,

further comprising: